Course: WebDev – Django Start.
Homework - 1

Theme: Install instruments for work and basic
Level: beginner
Instructor: Mikhail Nakonechnyi
Due Date: 15^{th} February, 2020

Problem 1
Consider the scalar system

$$\dot{x} = -x + u + w$$

w is zero-mean process noise with a variance of Q. The control has a mean value of u_0, an uncertainty of 2 (one standard deviation), and is uncorrelated with w. Rewrite the system equations to obtain an equivalent system with a normalized control that is perfectly known. What is the variance of the new process noise term in the transformed system equation?

Solution: The variance of the new process noise, w_u is $\Sigma_{w_u} = Q + \sigma_u^2 = Q + 4$.

$$\dot{x} = -x + u_0 + \underbrace{w + \Delta u}_{w_u}, \quad w_u \sim (0, Q + \sigma_u^2).$$

Problem 2
Consider the system

$$\begin{aligned} x_{k+1} &= \phi x_k + w_k, \\ y_k &= x_k, \end{aligned}$$

where $w_k \sim (0, 1)$, and $\phi = 0.9$ is an unknown constant. Design an extended Kalman filter to estimate ϕ. Simulate the filter for 100 time steps with $x_0 = 1, P_0 = I, \hat{x}_0 = 0$, and $\hat{\phi}_0 = 0$. Hand in your source code and a plot showing $\hat{\phi}$ as a function of time.

Solution: Perform the measurement update of the state estimate and estimation error covariance as follows

$$
\begin{aligned}
K_k &= P_k^- H_k^\top (H_k P_k^- H_k^\top + R_k)^{-1} = P_k^- H_k^\top (H_k P_k^- H_k^\top)^{-1}, \quad \text{Since } R_k = 0, \\
\hat{\bar{x}}_k^+ &= \hat{\bar{x}}_k^- + K_k(y_k - h_k(\hat{\bar{x}}_k^-, 0)) \\
&= \hat{\bar{x}}_k^- + K_k(y_k - \hat{x}_k^-), \quad \text{Since } \hat{\phi}_k^- = 0, \\
P_k^+ &= (I - K_k H_k) P_k^-
\end{aligned}
$$

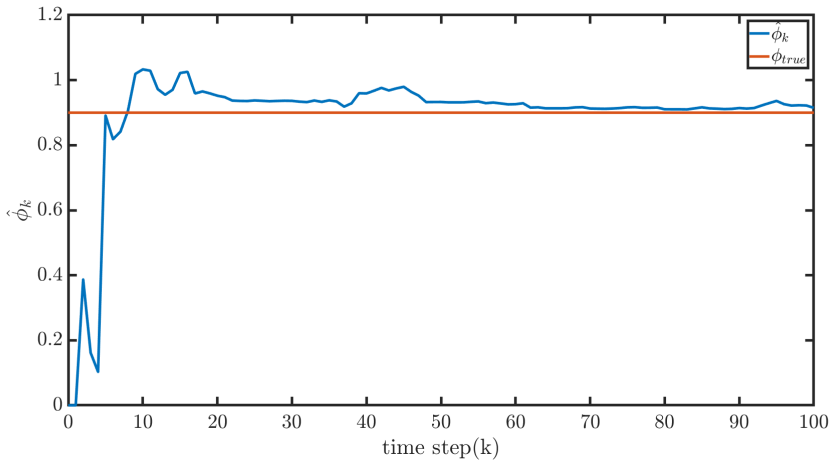

Figure 1: Plot showing $\hat{\phi}$ as a function of time.

```
%
% Venkatraman Renganathan
% MECH 6325 – Optimal Estimation & Kalman Filtering, Term: Fall 2019
% MATLAB Code for HW6Q2
%
clear; clc; close all;

%% Problem Data
H   = [1 0]; % Output Matrix
Qw  = 1;     % Process Noise Covariance
Q   = [Qw 0; % Augmented Process Noise Covariance diag(Qw,Q_phi),Q_phi = 0
       0 0];
N   = 100;   % Simulation Time Steps

% Initial values
x       = 1;
Pplus   = eye(2);
phiTrue = 0.9;
xHat    = 0;
phiHat  = 0;
varPhi  = [phiHat];
% Simulate the Discrete-Time EKF
```

```
for i = 1:N
    x      = phiTrue*x + sqrt(Qw).*randn; % x = phi*x + w, w~(0,1)
    y      = x;
    F      = [phiHat xHat;
               0      1];
    Pminus = F*Pplus*F' + Q;
    xHat   = phiHat*xHat;
    K      = Pminus*H'*inv(H*Pminus*H');
    z      = [xHat; phiHat];
    z      = z + K*(y - xHat);
    xHat   = z(1);
    phiHat = z(2);
    Pplus  = (eye(2) - K*H)*Pminus;
    % Store Value of phiHat
    varPhi = [varPhi phiHat];
end
% Plot the results
figure;
timeVec = 0:N;
plot(timeVec, varPhi);
hold on;
plot(timeVec, phiTrue*ones(N+1,1));
set(gca,'FontSize',12); set(gcf,'Color','White'); set(gca,'Box','on');
xlabel('time step(k)','interpreter','latex');
ylabel('$\hat{\phi}_{k}$','interpreter','latex');
legend('$\hat{\phi}_{k}$', '$\phi_{true}$','interpreter','latex');
a = findobj(gcf, 'type', 'axes');
h = findobj(gcf, 'type', 'line');
set(h, 'linewidth', 4);
set(a, 'linewidth', 4);
set(a, 'FontSize', 30);
set(gca,'TickLabelInterpreter','latex');
```